

# ANDROID NÂNG CAO

## BÀI 7: Sensors + Public Apps + AdMob

# Nội dung

## 1. Cảm biến (sensor)

- SensorManager
- Các loại sensors
- Viết ứng dụng La bàn

## 2. Các bước để public app

- Chữ ký số
- Public app

## 3. Quảng cáo với AdMob

## 4. Project cuối môn

Phần 1

# Cảm biến (sensor)

# SensorManager

- **Sensor:** chip cảm ứng nằm trong thiết bị, cung cấp các thông số cho Android OS
- **Quản trị chung bởi SensorManager**

```
SensorManager sm = (SensorManager)  
                    getSystemService(SENSOR_SERVICE);
```

- **Thông qua SensorManager ta có thể**
  - Lấy danh sách các sensor có trong hệ thống hiện tại
  - Lấy các đối tượng để làm việc trực tiếp với từng sensor
  - Đăng kí các listener để xử lý sự kiện do các sensor báo về

# SensorManager

## Các sensor hiện được Android OS hỗ trợ (class Sensor)

- TYPE\_ACCELEROMETER: cảm biến gia tốc
- TYPE\_AMBIENT\_TEMPERATURE: cảm biến nhiệt độ môi trường
- TYPE\_GRAVITY: cảm biến trọng lực
- TYPE\_GYROSCOPE: cảm biến con quay hồi chuyển
- TYPE\_LIGHT: cảm biến ánh sáng
- TYPE\_LINEAR\_ACCELERATION: cảm biến gia tốc tuyến tính
- TYPE\_MAGNETIC\_FIELD: cảm biến từ tính
- TYPE\_PRESSURE: cảm biến áp suất
- TYPE\_PROXIMITY: cảm biến khoảng cách gần
- TYPE\_RELATIVE\_HUMIDITY: cảm biến độ ẩm
- TYPE\_ROTATION\_VECTOR: cảm biến xoay
- TYPE\_GAME\_ROTATION\_VECTOR: cảm biến xoay 2D
- TYPE\_SIGNIFICANT\_MOTION: cảm biến chuyển động

# SensorManager

```
public class SensorActivity extends Activity
    implements SensorEventListener {

    final SensorManager sm;
    final Sensor light;

    public SensorActivity() {
        sm = (SensorManager) getSystemService(SENSOR_SERVICE);
        light = sm.getDefaultSensor(Sensor.TYPE_LIGHT);
    }
    protected void onResume() {
        super.onResume();
        sm.registerListener(this, light,
            SensorManager.SENSOR_DELAY_NORMAL);
    }
}
```

# SensorManager

```
protected void onPause() {
    super.onPause();
    sm.unregisterListener(this);
}

// khi độ chính xác của sensor thay đổi
// UNRELIABLE <-> LOW <-> MEDIUM <-> HIGH
public void onAccuracyChanged(Sensor s, int accuracy) {
}

// khi thông số sensor cập nhật
public void onSensorChanged(SensorEvent event) {
}
}
```

# Các loại sensor

- **Cảm biến trong Android chia làm 3 nhóm**
  - Cảm biến chuyển động
  - Cảm biến vị trí
  - Cảm biến môi trường
- **Mỗi loại sensor có những đặc điểm vật lý khác nhau, muốn hiểu chính xác các chi tiết các sensor cần đọc tài liệu hướng dẫn (cần có kiến thức nhất định về vật lý)**
- **Một số sensor là loại virtual (ảo), tức là kết quả được tính toán hoặc nội suy từ nguồn khác**



# Các loại sensor

- Android SDK không có các class định sẵn cho từng loại sensor mà chỉ có TYPE của sensor, dữ liệu do sensor trả về là **float** (trường hợp cảm biến 1 đầu ra – chẳng hạn đo ánh sáng) hoặc **float[]** (trường hợp cảm biến nhiều đầu ra)
- **TYPE\_AMBIENT\_TEMPERATURE**: cảm biến nhiệt độ, đơn vị đo là  $^{\circ}\text{C}$  (dễ nhất)
- **TYPE\_LIGHT**: cảm biến ánh sáng, đơn vị đo là lx
- **TYPE\_PRESSURE**: cảm biến áp suất không khí, đơn vị đo là mbar

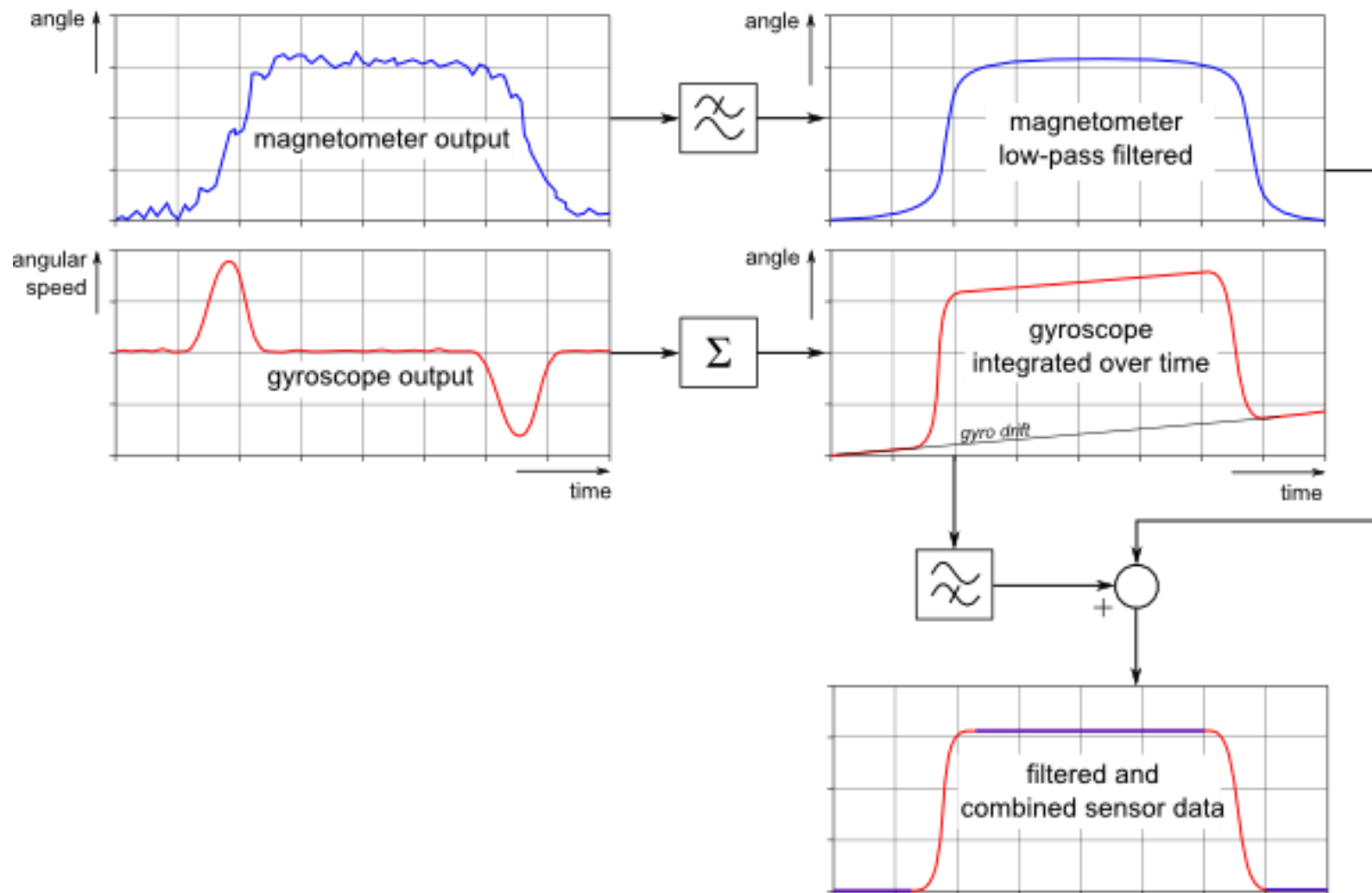
# Các loại sensor

- **TYPE\_PROXIMITY**: cảm biến khoảng cách đến đối tượng, đơn vị đo là cm
- **TYPE\_RELATIVE\_HUMIDITY**: cảm biến độ ẩm, đơn vị là %
- **TYPE\_ACCELEROMETER / TYPE\_GRAVITY**: cảm biến gia tốc / hấp dẫn trong 3D(x,y,z), đơn vị  $m/s^2$
- **TYPE\_GYROSCOPE**: cảm biến tốc độ góc quay trong 3D, đơn vị rad/s
- **TYPE\_MAGNETIC\_FIELD**: cảm biến lực từ trong 3D, đơn vị là  $\mu T$

# Kinh nghiệm khi làm với sensor

- **Nhất thiết phải giải phóng sensor khi không cần thiết, nếu không ứng dụng sẽ rất hao pin**
- **Hệ thống không tự động tắt sensor kể cả khi tắt màn hình**
- **Chú ý khi làm việc với các thông số 3D: các chiều có thể bị hoán đổi vị trí khi người sử dụng đặt thiết bị theo chiều âm (ví dụ: máy bị lật úp)**
- **Nên kiểm thử trên thiết bị thật và hiệu chỉnh độ nhạy dần dần**
- **Kết hợp nhiều sensor để thiết bị “nhạy cảm” hơn**

# Kinh nghiệm khi làm với sensor



# Ví dụ: La bàn đơn giản

```
public class MyCompassView extends View {
    private Paint area;
    private float arc = 0;
    // constructor
    public MyCompassView(Context context) {
        super(context);
        init();
    }
    // vẽ lại la bàn với góc mới
    public void updateData(float position) {
        arc = position;
        invalidate();
    }
}
```

# Ví dụ: La bàn đơn giản

// vẽ lại hình ảnh la bàn ứng với số liệu mới

```
protected void onDraw(Canvas c) {  
    int xPoint = getMeasuredWidth() / 2;  
    int yPoint = getMeasuredHeight() / 2;  
    float radius = (float) (Math.max(xPoint, yPoint) * 0.6);  
    c.drawCircle(xPoint, yPoint, radius, area);  
    float x = (float) (xPoint +  
        radius * Math.sin((double) (-arc) / 180 * Math.PI));  
    float y = (float) (yPoint -  
        radius * Math.cos((double) (-arc) / 180 * Math.PI));  
    c.drawLine(xPoint, yPoint, x, y, area);  
    c.drawText(String.valueOf(arc), xPoint, yPoint, area);  
}
```

# Ví dụ: La bàn đơn giản

// hàm vẽ lại hình ảnh của view

```
private void init() {  
    area = new Paint();  
    area.setAntiAlias(true);  
    area.setColor(Color.RED);  
    area.setStrokeWidth(3);  
    area.setStyle(Paint.Style.STROKE);  
    area.setTextSize(30);  
}  
}
```

# Ví dụ: La bàn đơn giản

```
public class MainActivity extends Activity {  
    private static SensorManager sensorService;  
    private MyCompassView compass;  
    private Sensor sensor = null;  
  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
        if (sensor != null)  
            sensorService.unregisterListener(mySensorEventListener);  
    }  
}
```



# Ví dụ: La bàn đơn giản

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    compass = new MyCompassView(this);
    setContentView(compass);
    sensorService = (SensorManager)
        getSystemService(Context.SENSOR_SERVICE);
    sensor = sensorService.getDefaultSensor(Sensor.TYPE_ORIENTATION);
    if (sensor != null)
        sensorService.registerListener(mySensorEventListener, sensor,
            SensorManager.SENSOR_DELAY_NORMAL);
    else
        finish();
}
```

# Ví dụ: La bàn đơn giản

```
private SensorEventListener mySensorEventListener =
    new SensorEventListener() {
        @Override
        public void onAccuracyChanged(Sensor s, int accuracy) {
        }
        @Override
        public void onSensorChanged(SensorEvent event) {
            compass.updateData(event.values[0]);
        }
    };
}
```

Phần 2

# Các bước để public app

# Chữ ký số

- **Khóa và chữ ký số: các ứng dụng android đều phải được xác nhận chủ sở hữu trước khi xuất bản lên các chợ ứng dụng**
- **Lý do cần xác nhận chủ sở hữu**
  - Đảm bảo việc ứng dụng không bị làm giả
  - Chủ sở hữu chịu trách nhiệm khi ứng dụng gây thiệt hại
- **Khi viết và chạy thử: sử dụng một khóa dành cho việc kiểm thử (Windows -> Preferences -> Android -> Build -> debug.keystore)**

# Chữ ký số

- Khi muốn xuất bản ứng dụng, người ta sẽ tự tạo một key riêng và giữ bí mật (để đảm bảo không ai có thể làm giả)
- Tạo key riêng sử dụng công cụ: keytool
- Hoặc có thể nhờ eclipse tạo hộ (sau đó phải lưu trữ lại, phòng khi nâng cấp ứng dụng)

# Public apps

- **Bước 1: đăng ký tài khoản Developer**  
**(<https://play.google.com/apps/publish/>)**
  - Có phí (25 USD/năm)
  - Thu nhập có thể đến từ:
    - Bán ứng dụng
    - Bán đồ trong ứng dụng
    - Bán quảng cáo
- **Bước 2: chuẩn bị thông tin về ứng dụng muốn xuất bản (mô tả - tiếng Anh là tốt nhất, các từ khóa liên quan, một số ảnh chụp màn hình)**

# Public apps

- **Bước 3: tạo ứng dụng dạng release**
  - Xóa bản dịch hiện tại: Project -> Clean
  - Xuất bản ứng dụng:
    - File -> Export -> Android -> Export Android Application
    - Chọn Project muốn xuất bản
    - Chọn key: dùng key đã có hoặc tạo mới
    - Nhập thông tin cần thiết cho chữ ký
    - Chọn nơi lưu file APK được xuất bản
- **Bước 4: upload ứng dụng**
- **Bước 5: chỉnh sửa nếu không được accept**

Phần 3

# Quảng cáo với AdMob



# AdMob

- Dịch vụ của Google, cho phép đặt ô quảng cáo vào ứng dụng, mỗi khi người dùng bấm vào quảng cáo, chủ sở hữu ứng dụng được chia một phần tiền mua quảng cáo
- Có nhiều dịch vụ quảng cáo khác cạnh tranh
- Có nhiều hình thức khác như hiện nguyên trang quảng cáo hoặc hiện trả tiền theo số lần tải ứng dụng,...

# AdMob

- Đăng ký AdMod (<https://www.admob.com>)
- Xem [developers.google.com/mobile-ads-sdk/](https://developers.google.com/mobile-ads-sdk/)

- Phải cấp quyền truy cập INTERNET cho APP

```
<uses-permission android:name="android.permission.INTERNET"/>
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

- Thêm thông tin vào AndroidManifest.xml

```
<activity android:name="com.google.android.gms.ads.AdActivity" />
```

```
<meta-data android:name="com.google.android.gms.version"
```

```
    android:value="@integer/google_play_services_version"/>
```

- Thêm thư viện Google Play Services

# AdMob

- Import `com.google.android.gms.ads.*`
- Thêm một `AdView` vào ứng dụng:

– Tạo `AdView`:

```
adView = new AdView(this);  
adView.setAdUnitId(MY_AD_UNIT_ID);  
adView.setAdSize(AdSize.BANNER);
```

– Thêm vào layout và nạp quảng cáo:

```
LinearLayout layout = (LinearLayout) findViewById(R.id.mainLayout);  
layout.addView(adView);  
AdRequest adRequest = new AdRequest.Builder().build();  
adView.loadAd(adRequest);
```

# Khớp hoạt động quảng cáo với app

```
public void onPause() {  
    adView.pause();  
    super.onPause();  
}
```

```
public void onResume() {  
    super.onResume();  
    adView.resume();  
}
```

```
public void onDestroy() {  
    adView.destroy();  
    super.onDestroy();  
}
```

Phần 4

# Project cuối môn

# Project cuối môn

- Project làm bởi 1-2 học viên
- Chủ đề xoay quanh các kiến thức được học
- Gặp giáo viên hướng dẫn hàng tuần để báo cáo tiến độ
- Kết quả:
  - Ứng dụng (phải được accept bởi Google Market)
  - Tài liệu thiết kế (có mẫu đi kèm)
  - Bảo vệ đồ án trước hội đồng

# Một số project tiêu biểu

- **Game hỗ trợ học tập:**
  - Học ngoại ngữ bằng hình ảnh, âm thanh
  - Luyện ngoại ngữ bằng trí nhớ
  - Học đếm và chữ cái cho trẻ em
- **Cung cấp thông tin:**
  - Đọc tin từ nhiều nguồn RSS
  - Món ăn và địa điểm
  - Truyện cổ tích cho trẻ em
- **Quản lý tài chính cá nhân**
- **Quản lý file trên Android**
- **Quản lý cấu hình mạng tự động**
- **Notification các sự kiện thể thao**
- **Gửi tin nhắn theo mẫu**
- **Báo thức và nhắc việc**